Some Complexity Aspects of the Control of Mobile Robots*

Magnus Egerstedt

magnus@ece.gatech.edu
Electrical and Computer Engineering
Georgia Institute of Technology
Atlanta, GA 30332

Abstract

In an earlier work it was shown how the length of the specification of a control procedure is affected by the availability of sensory information. In particular, it was shown that this length can be reduced by a factor that depends on the ratio of the size of the entire state space to the size of the set of states for which feedback is locally effective. In this paper we modify this result to explain why landmark-based navigation through a series of intermediary goals can be beneficial from a complexity point of view. We furthermore show how to choose the resolution of the sensors, i.e. the size of the output space, in order to generate control procedures with short description lengths.

1 Introduction

In this paper we continue the development begun in [5] of understanding how the choice of inputs to a dynamical system, e.g. a mobile robot, affects the length of the specification of the control procedure. In particular, in [5] we saw that when it is possible to use feedback in the specification the length can be reduced by a factor that depends on the ratio of the size of the entire state space to the size of the set of states for which feedback is locally effective. This result shows how to generate control procedures, with short description lengths, that drives the dynamical system between given boundary states. However, goals are seldom final goals. More often they tend to be intermediary goals in a grander scheme, which for instance is the case when mobile robots are navigating using landmarks. This paper modifies the results in [5] so that this can be taken into account in a systematic way by designing a controller with short description length that drives the system between intermediary goal states.

The second topic along these lines that we investigate in this paper is how to choose the resolution and the scope of the range sensors in order to manage the task successfully at the same time as we keep the description lengths short. It is clear that the size of the output space, i.e. the number of observations we make, affect the specification lengths of the control procedures. It would thus be desirable to come up with a theory that prescribes just how many observations should be made, e.g. a theory that tells us why ultrasonic sensors are to prefer over laser scanners in some applications, and vice versa.

To search for short descriptions of control procedures is a novel enterprise with a broad range of potential applications. For instance, in teleoperated robotics, the control signals are transmitted over communication channels in which the presence of channel noise makes it preferable to transmit instructions that are as short as possible. A related problem arises in the area of minimum attention control, where an attention functional is defined as a measure of the control variability. (See for example [2].) The problem then becomes that of minimizing the cost functional under the additional constraint that the servomechanism should perform in a satisfactory way. It can also be argued that this way of imposing complexity measures on control procedures has implications for decentralized or embedded control strategies, where the idea is to minimize the communication between different control modules at the same time as sufficient information must be available in order for the overall system to meet its specifications.

*Funded in part by the Georgia Tech 2100129 Yamacraw grant.

1.1 Free-Running, Feedback Automata

In order to formalize the notion of specification complexity we focus our attention on the finitely describable aspects of finite state machines. The input symbols to our finite state machines will be drawn from the finite set S, and we use $S^\star$ to denote the set of all words over S, including the empty one. We let $s \in S$ denote an element in S, and use boldface $\mathbf{s} \in S^\star$ to denote elements in $S^\star$. If we define the associative operation of concatenation on $S^\star$, the empty word serves as an identity under this operation. Thus $S^\star$ is the *free monoid* generated by S. Now, consider the finite sets S, U. We will let S^U denote the set of mappings from U to S, and we use $card(S)$ for the cardinality of S.

If we let X, V be finite sets, and let $\delta \in X^{X\times V}$, then we can identify (X, V, δ) with a *finite automaton* (see for example [1, 6]), whose operation is given by

$$x_{k+1} = \delta(x_k, v_k).$$

If we add another finite set Y and a mapping $\gamma \in Y^X$ to the definition, we get an *output* automaton $(X, Y, V, \delta, \gamma)$, where $x_{k+1} = \delta(x_k, v_k)$ and $y_k = \gamma(x_k)$.

Given a word $\mathbf{v} \in V^\star$, where $\mathbf{v} = v_1 \cdots v_p$, we use $\delta(x, \mathbf{v})$ as shorthand for

$$\delta(\delta(\cdots(\delta(x, v_1), v_2)\cdots, v_{p-1}), v_p),$$

and we let v^p denote the word obtained by concatenating v with itself $p-1$ times, i.e. $v^p = v\cdots v$.

We now introduce the main object of study, i.e. a dynamical system called a *free-running, feedback automaton*, as defined in [5]. The idea is to let such an automaton read an input from a given alphabet, and then advance the state of the automaton repeatedly (free-running property) without reading any new inputs until an interrupt is triggered. We furthermore want to impose additional structure on the input set to allow for feedback signals to be used. Hence a FRF-automaton is a free-running automaton whose input alphabet admits the structure $\Sigma = V \times K \times \mathcal{T}$, where V is a finite set, $K \subset V^{Y\times V}$, and $\mathcal{T} \subset \{0,1\}^Y$. Thus the input to a FRF-automaton is a triple (v, κ, τ), where $v \in V, \kappa : Y \times V \to V$, and $\tau : Y \to \{0, 1\}$.

Definition 1.1 *(Free-Running, Feedback Automaton) Let X, Y be finite sets and let $\delta : X \times V \to X$, $\gamma : X \to Y$ be given functions. Let $\Sigma = V \times K \times \mathcal{T}$, where V is a finite set, $K \subset V^{Y\times V}$, and $\mathcal{T} \subset \{0,1\}^Y$. We say that $(X, \Sigma, Y, \delta, \gamma)$ is a free-running, feedback automaton whose evolution equation is*

$$x_{k+1} = \delta(x_k, \kappa_{l_k}(\gamma(x_k), v_{l_k})),$$

where

$$l_{k+1} = \begin{cases} l_k \ \textit{if}\ \tau_{l_k}(\gamma(x_{k+1})) = 0 \\ l_k + 1 \ \textit{otherwise,} \end{cases}$$

given the input string $(v_1, \kappa_1, \tau_1)\cdots(v_p, \kappa_p, \tau_p) \in \Sigma^\star$.

It should be noted that the free-running property of the FRF-automata implies that they can, in general, be guided along a path using fewer instructions than the classical finite automata. However, since the input set to a finite automaton is a finite set V, while the input set to the corresponding FRF-automaton is of the form $V \times K \times \mathcal{T}$, where $K \subset V^{Y\times V}, \mathcal{T} \subset \{0,1\}^Y$, the input set has a higher cardinality in the latter of these cases. Any reasonable measure of the complexity of a control procedure must take the size of the input space into account since the number of bits required to code a word over a given alphabet typically depends logarithmically on the size of the alphabet. (See for example [4].) This dependency is captured in a natural way if we define the specification complexity of a control procedure as the description length of the input sequence.

Definition 1.2 *(Description Length) Consider a finite set S. We say that a word $\mathbf{s} \in S^\star$ has description length*

$$\mathcal{D}(\mathbf{s}, S) = |\mathbf{s}| \log_2(card(S)),$$

where $|\mathbf{s}|$ is the length of $\mathbf{s}$.

Definition 1.3 *(Specification Complexity) Consider a FRF-automaton, A, with state space X and input set Σ. Let $\boldsymbol{\sigma}$ be the word of minimal description length over Σ that drives the automaton between two given states $x_0, x_f \in X$. We then say that the task of driving A between x_0 and x_f has specification complexity $\mathcal{C}(A, x_0, x_f) = \mathcal{D}(\boldsymbol{\sigma}, \Sigma)$.*

2 Specification Complexity

We here review the main result from [5] in order to see how the complexity of the instructions can be reduced when using landmark-based navigation. Since the cardinality of the input set depends on the size of the domain of the feedback mapping, a smaller domain can be expected to reduce the complexity. In order to make this observation rigorous, we need to introduce the notions of ballistic reachability and control-invariant reachability: A set $X_s \subset X$ is *ballistically reachable from* x if there exists a $v \in V$ such that $\delta(x, v^q) \in X_s$ for some $q \in \mathbb{Z}^+$. Furthermore, X_s is ballistically reachable from $X_t \subset X$ if there exists a $v \in V$ such that for all $x \in X_t$ it holds that

$\delta(x, v^{q(x)}) \in X_s$ for some $q(x) \in \mathbb{Z}^+$. An element $x \in X_s \subset X$ is said to be *control-invariantly reachable in* X_s if it can be reached from all states in X_s without the trajectory leaving X_s.

Now, in order to compare purely open-loop control, i.e. when no observations are made, with a situation where sensory information is available we must be able to generate open-loop motions on the FRF-automata. It is clear that the input sequence $\boldsymbol{\sigma}_{ol} = (v_1, \kappa_{ol}, \tau_{ol}) \cdots (v_q, \kappa_{ol}, \tau_{ol}) \in \Sigma^\star$, where $\kappa_{ol}(v, y) = v \ \forall v \in V,\ y \in Y,\ \tau_{ol}(y) = 1 \ \forall y \in Y$ achieves this. However, this word has length q, and it is drawn from the input alphabet $\Sigma = V \times V^{Y \times V} \times \{0,1\}^Y$, and thus the description length is $\mathcal{D}(\boldsymbol{\sigma}_{ol}, \Sigma) = q \log_2(card(\Sigma))$. But, this is clearly not the result we would like to have. Instead we can restrict the input alphabet to be $\Sigma_{ol} = V \times \{\kappa_{ol}\} \times \{\tau_{ol}\}$, which has cardinality $card(V)$. The description length of $\boldsymbol{\sigma}_{ol}$ is now $\mathcal{D}(\boldsymbol{\sigma}_{ol}, \Sigma_{ol}) = q \log_2(card(V))$, relative to the smaller input set Σ_{ol}.

Consider the connected, classical, finite automaton $A = (X, V, \delta)$. We recall that the *backwards eccentricity* of a state, $ecc(A, x)$, denotes the minimum number of instructions necessary for driving the automaton from any other state to x. (See for example [3].) We furthermore let the *radius* of A be given by

$$radius(A) = \min_{x \in X} ecc(A, x).$$

Now consider the FRF-automaton $\tilde{A}$. If we let

$$\mathcal{C}(\tilde{A}, x) = \max_{x_0 \in X} \mathcal{C}(\tilde{A}, x_0, x),$$

then we directly get that

$$\mathcal{C}(A_{ol}, x) \geq radius(A) \log_2(card(V)),$$

where A_{ol} is the FRF-automaton $(X, Y, \Sigma_{ol}, \delta, \gamma)$, and A is the classical automaton (X, V, δ).

Theorem 2.1 *(Egerstedt and Brockett [5]) Assume that* $card(V) \geq 2$. *Suppose that* $x_f \in X_f$, *where* X_f *is an observable subset for the finite automaton* A, *i.e. it is possible to construct an observer that converges in a finite number of steps on the subset* X_f. *Assume that* $card(\gamma(X_f)) < card(X_f)$ *and* $\gamma(X_f) \cap \gamma(X \backslash X_f) = \emptyset$. *If* X_f *is ballistically reachable from* $X \backslash X_f$, *and* x_f *is control-invariantly reachable in* X_f, *then there exists a FRF-automaton* $A_{FRF} = (X, Y, \Sigma', \delta, \gamma)$ *such that*

$$\frac{\mathcal{C}(A_{FRF}, x_f)}{\mathcal{C}(A_{ol}, x_f)} \leq \frac{4 card(X_f)}{radius(A)}.$$

3 Navigation Using Landmarks

It is clear that the premise on which the previous theorem is based is too restrictive to capture the desired *chained* structure that intermediary goals give rise to. Instead we need to extend the trajectories from Theorem 2.1 through a chain of goal states. This can be achieved by assuming that we work with an automaton where subset-observers can be designed around different states, i.e. the intermediate goals. We also assume that the sets on which the observers are defined are ballistically reachable from each other. We could then use open loop control for driving the system between these sets on the parts of the state space where the lack of sensory information prevents effective use of feedback. We compliment this with feedback controllers on the subsets where subset-observers can be constructed.

However, before we are ready to formulate this as a theorem, some comments about subset-observers should be made. Consider the finite automaton $(X, Y, V, \delta, \gamma)$. We define the *output sequence map* $\mathcal{O} : \mathbb{Z}^+ \times X \times V^Y \to Y^\star$ as

$$\mathcal{O}(p, x, w) = \gamma(x_1) \cdot \gamma(x_2) \cdots \gamma(x_p),$$

where $w : Y \to V$, and $x_1 = x$, $x_2 = \delta(x_1, w(\gamma(x_1))), \ldots, x_p = \delta(x_{p-1}, w(\gamma(x_{p-1})))$. Note that in the output sequence map $y_1 \cdot y_2$ denotes the concatenation of the letters y_1 and y_2 from the finite alphabet Y, and $\mathcal{O}(p, x, w) \in Y^p \subset Y^\star$, where Y^p is the set of words of length p over Y.

Definition 3.1 *(Observable Subset) Consider the finite automaton* $(X, Y, V, \delta, \gamma)$. *A subset* $X_g \subset X$ *such that* $\gamma(X_g) \cap \gamma(X \backslash X_g) = \emptyset$ *is said to be observable if there exist a positive integer* p_{obs} *and a* $w_{obs} : Y \to V$ *that satisfies the following conditions:*

- $\mathcal{O}(p_{obs}, x_1, w_{obs}) \neq \mathcal{O}(p_{obs}, x_2, w_{obs}),\ \forall x_1, x_2 \in X_g, x_1 \neq x_2$;
- *For all* $x_1 \in X_g$ *it follows that* $x_q \in X_g,\ q = 1, \ldots, p_{obs}$, *where* $x_2 = \delta(x_1, w_{obs}(\gamma(x_1))),\ x_3 = \delta(x_2, w_{obs}(\gamma(x_2))), \ldots$

Definition 3.2 *(Subset-Observer Automaton) Consider the finite automaton* $A = (X, Y, V, \delta, \gamma)$, *where* $X_g \subset X$ *is an observable subset.* (Z, O, Ω, g, h), *where* Z, O *are finite sets,* $\Omega = V \times V^{O \times V}$, $g : Z \times Y \times \Omega \to Z$, *and* $h : Z \times Y \to O$ *is a subset-observer automaton to* A *if there exists a* $\omega = (v, w) \in \Omega$ *such that the following conditions hold:*

$$x_{k+1} = \delta(x_k, w(o_k, v)),\ y_k = \gamma(x_k)$$
$$z_{k+1} = g(z_k, y_k, w(o_k, v)),\ o_k = h(z_k, y_k)$$

gives that the current state in Z can be mapped uniquely to the current state in X after sufficiently many iterations. Also, for all $x_1 \in X_g$ it holds that $x_q \in X_g$, $q = 1, \ldots, p_{obs}$, where $x_2 = \delta(x_1, w(o_1, v))$, $x_3 = \delta(x_2, w(o_2, v))$, and so on.

From [5] we have the following result, presented in a form modified for the purpose of this paper:

Lemma 3.1 *(Subset-Observers [5]) Let $X_g \subset X$ be an observable subset to the finite automaton $A = (X, Y, V, \delta, \gamma)$. Then a subset-observer automaton (Z, O, Ω, g, h) to A can always be constructed with state space of cardinality less than or equal to $1 + card(\gamma(X_g)) + card(X_g)$.*

Theorem 3.1 *(Navigation Using Landmarks) Assume that $card(V) \geq 2$. Let the sets $X_1, \ldots, X_n$ be disjunct, observable subsets with cardinality less than or equal to C, where $card(\gamma(X_i)) < C$, $i = 1, \ldots, n$, $\gamma(X_i) \cap \gamma(X \backslash X_i) = \emptyset$, $\gamma(X_i) \cap \gamma(X_j) = \emptyset$, $i \neq j$. Let $x_f \in X_n$ be control-invariantly reachable in X_n and let X_1 be ballistically reachable from x_0. Assume that there exists intermediary goals $x_i \in X_i$, $i = 1, \ldots, n-1$ such that x_i is control-invariantly reachable in X_i and X_{i+1} is ballistically reachable from x_i. Then there exists a FRF-automaton $A_{FRF} = (X, Y, \Sigma', \delta, \gamma)$ such that*

$$\frac{\mathcal{C}(A_{FRF}, x_f)}{\mathcal{C}(A_{ol}, x_f)} \leq \frac{4nC}{radius(A)}.$$

Proof: The proof is based on a combination of the proofs of Theorem 2.1 and Lemma 3.1 respectively, that can be found in [5]. We let $\Sigma' = \{\hat{v}\} \times V^O \times \{0,1\}^O$, where O is the state-space of the observer obtained by combining the subset-observers defined on the different observable subsets, and where $\hat{v}$ is any arbitrary $\hat{v} \in V$. It is already shown in [5] that it is possible to drive this FRF-automaton from any initial state to x_f using only one input.

An upper bound on the size of the input space can be derived from Lemma 3.1 as

$$\begin{aligned} card(\Sigma') &\leq (2card(V))^{\sum_{i=1}^{n}(1+card(\gamma(X_i))+card(X_i))} \\ &\leq card(V)^{4nC}. \end{aligned}$$

Now, since $\mathcal{C}(A_{ol}, x_f) \geq radius(A) \log_2(card(V))$, we have

$$\frac{\mathcal{C}(A_{FRF}, x_f)}{\mathcal{C}(A_{ol}, x_f)} \leq \frac{4nC}{radius(A)}.$$

The theorem thus follows. ∎

One conclusion to be drawn from Theorem 3.1 is that the increase in description length, caused by the summation over many intermediate goals, can be counter-acted by making the sets where feedback is effective small. In the mobile robot case, this would correspond to using many easily detectable landmarks as a basis for the navigation system.

4 Sensor Selection

Another issue that can be tackled within the specification complexity context is how sensor selection for mobile robots affects the description lengths of the control procedures. In other words, we want to be able to determine which observations to make. For this we assume that the state space is defined on a bounded lattice, which is reasonable in mobile robot applications.

4.1 Lattice Automata

Consider a lattice $\mathcal{L}$ in $\mathbb{Z}^d$ with the Manhattan metric defined on it, i.e. if $x_1, x_2 \in \mathcal{L}$ then $\mathcal{M}(x_1, x_2)$ is given by the number of transitions along edges that one needs to make in order to go from x_1 to x_2. If we assume that from a given node we allow transitions along all adjacent edges, then the input set U has cardinality $card(U) = 2d$, where $d \geq 2$ is the dimension of the lattice.

We now introduce the concept of a k-sensor. The idea is that if we have a k-sensor we can, from any state x_0, directly observe all states x such that $\mathcal{M}(x_0, x) \leq k$. The number of such states that we can observe are given by $2d^k$, if we exclude x_0 itself.

Let us now assume that the state space is given by a bounded lattice, $\mathcal{L}_B$, forming a hypercube in $\mathbb{Z}^d$, and that we have a method for distinguishing boundary states from non-boundary states in the lattice, i.e. we can determine whether a node is on the boundary of the lattice or not. Since we can observe all states of distance less than or equal to k from the goal with our k-sensor, we get the output space

$$Y_k = \{n, b, e_k, x_0, \ldots, x_{2d^k}\},$$

where n denotes "no boundary", b denotes "boundary", e_k denotes "not k-close to the goal", and the remaining outputs correspond to the $2d^k + 1$ states that are visible from the goal, as long as the goal state satisfies $\mathcal{M}(x_g, x_b) \geq k$ for all boundary states x_b.

Since we have already established that $card(U) = 2d$ our entire input space has cardinality

$$card(\Sigma) = 2d(2d)^{2d^k+4} 2^{2d^k+4},$$

and the associated FRF-automaton becomes:

Definition 4.1 *(Lattice Automaton) Let the automaton $A_{\mathcal{L}_B}$ be given by $(\mathcal{L}_B, \Sigma, Y_k, \delta, \gamma)$, where $\mathcal{L}_B$ is a hypercubical lattice, $\Sigma = U \times U^{Y_k} \times \{0,1\}^{Y_k}$, with $card(U) = 2d$. Furthermore, $\delta : \mathcal{L}_B \times U \to \mathcal{L}_B$ and $\gamma : \mathcal{L}_B \to Y_k$ are directly induced by the lattice structure of the state space.*

Since a bounded lattice is a *traceable graph* (see for example [4]), i.e. it admits a *Hamiltonian*, or *spanning path* that visits each node exactly once then the worst case number of instructions, necessary for intersecting a goal state at an unknown location, in the open-loop case, is $card(X)$. However, the question that needs to be answered is how many instructions suffice in the closed-loop, free-running case.

Lemma 4.1 *Given the FRF-automaton in Definition 4.1, with a goal point that satisfies $\mathcal{M}(x_g, x_b) \geq k$ for all boundary states x_b. Then it is possible to traverse the lattice and intersect the region containing the $2d^k + 1$ states around x_g using less than or equal to*

$$\left(\frac{2}{k}\right)^{d-1} card(X)^{(d-1)/d}$$

instructions. Furthermore, when the goal is visible it can be reached using only one instruction, i.e. in the closed-loop case we get an upper bound of $(2/k)^{d-1} card(X)^{(d-1)/d} + 1$ instructions.

Proof: Along each dimension, a total of $card(X)^{1/d}$ nodes can be visited. We first assume, without loss of generality, that we traverse a 2-dimensional layer of the state space, starting at the "south-western" corner. Then we can move along the y-axis using only one instruction, until a boundary state is intersected. We then move k steps "east" along the x-axis, and repeat this procedure until the entire 2-dimensional lattice-plane is traversed, giving us a total of $2(card(X)^{1/d})/k$ instructions.

Now, by repeating this along the remaining dimensions we directly get a total number of

$$\left[2\left(\frac{card(X)^{1/d}}{k}\right)\right]^{d-1}$$

instructions, and the lemma follows. ∎

It should be noted that since the proof of Lemma 4.1 is constructive, we have no guarantee that this is in fact the shortest procedure on the average. Instead it is merely a construction that serves as a tool when we discuss how to actually choose the value of k in our k-sensor.

4.2 Optimal Sensor Selection

In light of Lemma 4.1, the *complexity ratio* between the specification complexities associated with the free-running, closed-loop case and the purely open-loop case becomes

$$\frac{\left(\left(\frac{2}{k}\right)^{d-1} card(X)^{(d-1)/d} + 1\right)\left(1 + \log_2 d + (2d^k + 4)(2 + \log_2 d)\right)}{card(X)(1 + \log_2 d)}.$$

Now, let k take on values over the reals and assume that $card(X)$ and d are fixed. Then the second derivative of the numerator in the complexity ratio is of the form

$$C(k) + D(k)\left(\ln d - \frac{2}{k}\right),$$

where $C(k)$ and $D(k)$ are positive, which implies that $f(k)$ is convex as long as $k > 2/\ln d$, as seen in Figure 1, and we state this fact as a proposition.

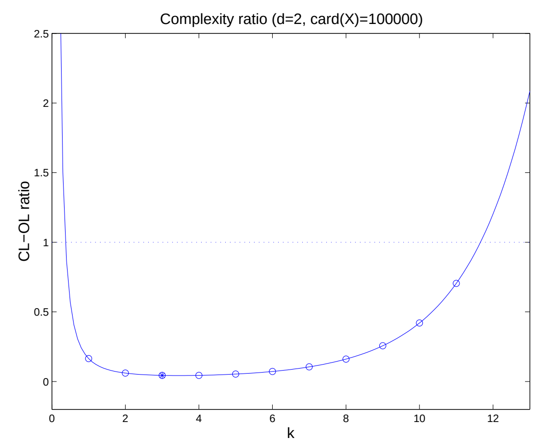

Figure 1: The ratio between the closed-loop and open-loop specification complexity is shown as a function of k.

Proposition 4.1 (Optimal Sensor Selection) *Given the FRF-automaton in Definition 4.1. Then there exists a (not necessarily unique) value $k^\star \in \mathbb{Z}^+$ such that $k^\star$ minimizes the complexity ratio.*

However, if we observe directional information instead of the actual states, i.e. we perform some preprocessing of the data, we can think of this as having a new output set $\tilde{Y}_k$, with cardinality $card(\tilde{Y}_k) = 2d + 3$, since we have $2d$ directions to the goal that we can observe, together with the three original outputs $\{e_k, b, n\}$. This can thus be thought of as adding a preprocessor to the sensor that performs a very structured data compression that only keeps angular but no radial information. Our new FRF-automaton thus becomes:

Definition 4.2 *Let the FRF-automaton $\tilde{A}_{\mathcal{L}_B}$ be given by $(\mathcal{L}_B, \Sigma, \tilde{Y}_k, \delta, \tilde{\gamma})$, where $\mathcal{L}_B$ and Σ are given in Definition 4.1, and $\tilde{Y}_k$ is given in the previous paragraph.*

In this "preprocessing" case the complexity ratio becomes

$$\frac{\left(\left(\frac{2}{k}\right)^{d-1} card(X)^{(d-1)/d}+1\right)(1+(2d+3)(2+\log_2 d)+\log_2 d)}{card(X)(1+\log_2 d)}.$$

The derivative of the numerator of this ratio is

$$-(d-1)\frac{(2card(X)^{1/d})^{d-1}}{k^d}F(d),$$

where $F(d)$ is positive for all d, which implies that the derivative is negative for all positive k. Hence the ratio is monotonously decreasing, which is consistent with Figure 2.

Proposition 4.2 *In the "preprocessing" case, i.e. given the FRF-automaton in Definition 4.2, the complexity ratio decreases monotonously in k.*

Remark 4.1 *What this implies is that in the preprocessing case, the stronger the sensor the better.*

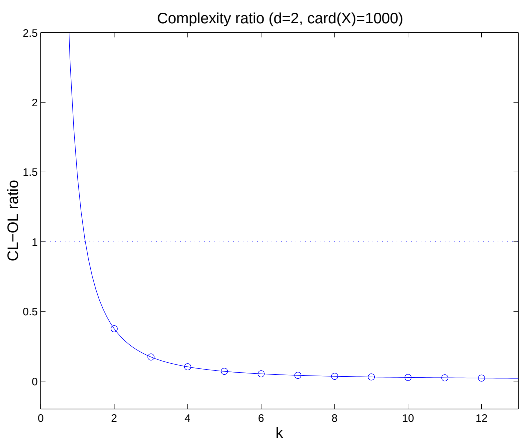

Figure 2: In this figure it is shown that in the preprocessing case, the complexity ratio decreases monotonously in k.

5 Conclusions

The results reported in this paper extends those in [5] in two specific areas. First of all, it is shown how the specification complexity of the control procedure for driving a mobile robot can be reduced if the robot traverses through a collection of intermediary goal states. This is the case when robots navigate using landmarks, or when directional instructions are based on easily recognizable waypoints.

Secondly, it is shown how it is possible to evaluate the performance of particular sensors (or more specifically, the size of the output alphabet) within this framework. This has implications to both how mobile robots should be controlled, as well as how to design these vehicles in terms of sensor selection. A natural extension of this result would be to investigate the choice of the input set U as well. This would thus provide a unified treatment of actuator and sensor selection, as well as control design.

References

[1] M. Arbib, Ed. *Algebraic Theory of Machines, Languages, and Semigroups*, Academic Press, New York, 1968.

[2] R.W. Brockett. Minimum Attention Control. Proceedings of the *IEEE Conference on Decision and Control*, pp. 2628–2632, 1997.

[3] F. Buckley and F. Harary. *Distance in Graphs*. Addison-Wesley Publishing Company, New York, 1990.

[4] T.M. Cover and J.A. Thomas. *Elements of Information Theory*, John Wiley & Sons, Inc., New York, 1991.

[5] M. Egerstedt and R.W. Brockett. Feedback Can Reduce the Specification Complexity of Motor Programs. *IEEE Conference on Decision and Control*, Orlando, FL, Dec. 2001.

[6] A. Ginzburg. *Algebraic Theory of Automata*. ACM Monograph Series, Academic Press, New York, 1968.